



Obteniendo el Token #2

Transcripción

[00:00] Lo que sigue es tratar de hacer un handling a esa excepción que estamos recibiendo. ¿Cómo? Porque por ejemplo ya vimos que si el header no llega, me va a lanzar un null pointer exception. ¿Pero qué pasaría si el token es incorrecto o viene un string vacío o viene cualquier cosa?

[00:18] Entonces lo que yo voy a hacer aquí es hacer antes del replace, porque aquí si yo hago un replace a algo vacío todo me puede dar una excepción. Entonces voy a comentar esto aquí primero, punto y coma, porque lo que voy a hacer aquí, es que si el token es igual a vacío, entonces voy a hacer un `throw new RuntimeException`.

[00:50] Y listo. Y voy a poner aquí “El token enviado no es valido”. Y listo. El token está vacío. O si token está nulo. Entonces ahora voy a preguntar aquí, if token null o está vacío, entonces voy a lanzar un `new RuntimeException`. Y solamente si ya no llegó a esta línea voy a hacer el reemplazo de aquí: `token = token.replace`. Y listo.

[01:34] Con esto ya tenemos nuestra implementación ya saneada. Vamos solamente a probar, vamos a guardar. Limpiamos, corremos nuestro servidor. Siempre demora un poco en mi máquina pero listo, ya está. Vamos a limpiar y tenemos nuestra respuesta OK, y ya está todo funcionando correctamente y estamos imprimiendo el código como debe ser.

[02:03] Entonces siempre recuerden hacer el código fault tolerant como se le dice, pensando siempre en los corner cases que pueden suceder, si

estuviéramos trabajando con TDD por ejemplo sería muy fácil escribir nuestros tests, nuestros casos de uso cuando un token llega vacío, cuando llega null, y ya tendríamos cubiertos estos escenarios que pueden suceder potencialmente y no como ahora que lo hemos visto cuando sucedió recién en el momento de la implementación del código.

[02:34] Es un consejo. ¿Qué es lo que falta ahora? Bueno, como vimos que yo al token le puedo mandar datos de expiración, datos de quién lo está firmando y datos de quién está asignado a ese token, ahora lo que yo necesito hacer es la parte de validar si el token es válido, valga la redundancia, para liberar el acceso o bloquear el request, porque lo que yo puedo hacer aquí después preguntar algo así como `if tokenEsValido`, entonces yo puedo llamar aquí a `filterChain.doFilter`.

[03:16] Y si no, no hacer nada y simplemente el request va a quedar ahí. Eso vamos a ver en el siguiente video. Nos vemos.